

Domain Primitives compatibles avec le pattern V1

Cette liste recense les primitives métiers pour lesquelles la transposition du **pattern V1** (VO type par unite, avec ``UnitRegistry`` et ``convert(...)``) est naturelle, cohérente et recommandée.

Domain Primitives a transposition naturelle

Primitive Metier	Scalaire encapsule	Unite typable	Exemple usage	Conversion possible
-----	-----	-----	-----	-----
Distance	<code>`number`</code>	<code>`km`</code> , <code>`miles`</code> , <code>`meters`</code>	<code>`Distance(5, 'km')`</code>	km miles
Poids	<code>`number`</code>	<code>`kg`</code> , <code>`lb`</code> , <code>`g`</code>	<code>`Weight(70, 'kg')`</code>	kg lb g
Temperature	<code>`number`</code>	<code>`C`</code> , <code>`F`</code> , <code>`K`</code>	<code>`Temperature(20, 'C')`</code>	C F K
Duree	<code>`number`</code>	<code>`seconds`</code> , <code>`minutes`</code> , <code>`hours`</code>	<code>`Duration(90, 'minutes')`</code>	toutes
Monnaie	<code>`number`</code>	<code>`EUR`</code> , <code>`USD`</code> , <code>`JPY`</code>	<code>`Money(10, 'EUR')`</code>	si taux disponibles
Angle	<code>`number`</code>	<code>`degrees`</code> , <code>`radians`</code>	<code>`Angle(90, 'degrees')`</code>	radians degres
Energie	<code>`number`</code>	<code>`kWh`</code> , <code>`J`</code> , <code>`cal`</code>	<code>`Energy(5, 'kWh')`</code>	toutes
Vitesse	<code>`number`</code>	<code>`km/h`</code> , <code>`m/s`</code> , <code>`mph`</code>	<code>`Speed(100, 'km/h')`</code>	toutes
Volume	<code>`number`</code>	<code>`L`</code> , <code>`m3`</code> , <code>`gallon`</code>	<code>`Volume(3, 'L')`</code>	toutes
Tension electrique	<code>`number`</code>	<code>`V`</code> , <code>`mV`</code>	<code>`Voltage(230, 'V')`</code>	V mV
Frequence	<code>`number`</code>	<code>`Hz`</code> , <code>`kHz`</code> , <code>`MHz`</code>	<code>`Frequency(50, 'Hz')`</code>	Hz kHz MHz
Debit (reseau)	<code>`number`</code>	<code>`Mbps`</code> , <code>`Gbps`</code> , <code>`kB/s`</code>	<code>`Bandwidth(100, 'Mbps')`</code>	toutes
Pression	<code>`number`</code>	<code>`Pa`</code> , <code>`bar`</code> , <code>`psi`</code>	<code>`Pressure(1, 'bar')`</code>	toutes
Surface	<code>`number`</code>	<code>`m2`</code> , <code>`cm2`</code> , <code>`ft2`</code>	<code>`Area(40, 'm2')`</code>	toutes

Bonus : primitives enrichies compatibles

Primitive enrichie	Scalaire + contexte	Exemple
-----	-----	-----
Quantite avec unite	<code>`number`</code> + unite composite (<code>`boxes`</code> , <code>`items`</code>)	<code>`Quantity(10, 'boxes')`</code>
Score / Pourcentage	<code>`number`</code> + <code>`%`</code> ou sur 100	<code>`Score(85, '%')`</code>
Taux / Ratio	<code>`number`</code> + <code>`bps`</code> , <code>`percent`</code>	<code>`Rate(0.02, 'percent')`</code>

Criteres de selection

Ces Domain Primitives partagent les caracteristiques suivantes :

- Encapsulation dun scalaire (``number``)
- Unite metier clairement definie (``as const``)
- Logique de conversion reversible ou commutable
- Usage pedagogique et fonctionnel naturel
- Forte compatibilite avec ``UnitRegistry`` + ``convert(...)``

Ce document peut servir de reference pour des implementations pedagogiques ou des projets visant a appliquer les Value Objects types dans des domaines metiers.